Star Wars — The Movies Fan Page

Introduction

You are required to develop a website to showcase the Star Wars films along with some behind-the-scenes details. While the expectations about the functionality of this website are pretty basic — at the very minimum there are forms which are meant to retrieve and send data to a database — there's also a Search Engine Optimisation (SEO) component to this project. The core functionality along with the SEO steps in your solution will ensure a reasonable grade for this project. And then, to get the top marks you may implement some optional features, such as user registration and authentication, or including JavaScript charts to visualise the data.

There are websites who show all imaginable information about Star Wars; e.g.: StarWars.com (https://www.starwars.com/databank) and Wookieepedia (https://starwars.fandom.com/wiki/Main_Page) — the list all films and series made till today. The data used in the project, is limited to the first seven films made; links to images from those websites are included for improved presentation of the data and an improved aesthetic browsing/interaction experience. For inspiration and examples of design you may want to browse through StarWars.com and Wookieepedia.

StarWars (SQLite) Database

A small database with the first seven Star Wars films is provided to help you get started. There are 12 main tables with `film` in the centre of them all. You may need to add more tables or views if you feel this is necessary to implement specific features required for your project. Here's the description of the main tables in the `star_wars` database:

- `climate` — the climate of planets.
- `film` — the film details and description of the film, including the film poster/image URL.
- `manufacturer` — the manufacture of vehicles and spaceships with their symbol if known.
- `people` — physical details about the characters played in the film, including the image URL.
- `planet` — scientific details about planets and their appearance.
- `producer` — the producer of the film, along with the image URL.
- `species` — physical and background details about species and an image of how they look.
- `starship` — extra information about starships with an image.
- `starshipclass` — the class type of a starship.
- `terrain` — the terrain type of a planet with an example image from the film.
- `vehicle` — extra information about vehicles with an image.
- `vehicleclass` — the class type of a vehicle.

To get a better idea of the data go ahead and explore the databases in the PHPStorm Database/Query Console/Tool. It is essential that you examine the structure of the tables — this will give you a clear idea what data is expected when designing SQL insert queries and the pages for say, recording the `people` data. As explained in Tutorial 4, make a connection with the database `star_wars.db` and view the tables and columns in the PHPStorm Database Tool. Please note that while we are using SQLite for this project, which is a server-less database, a more realistic scenario would have a server-based database, e.g.: MySQL, PostgeSQL or any others from the same shelf.

Solution Requirements

The core task is to develop a website for showcasing and augmenting the Star Wars films information. This task will require a bit of front-end work for designing HTML pages with forms and back-end functionality written in PHP code; and then, to give the website a proper look-and-feel you'll need to add some interactivity with JavaScript and CSS. There is a lot of freedom as to how you may approach this task. At the very minimum your solution must include these basic features/properties:

- The start/home/index page for your solution, providing the user and search engine robots/crawlers with the basic information about your site. This page should also include the navigation menu for accessing the rest of the pages in your solution.
- The overview page to display all Star Wars films and possibility to view more details for any of the items. You should be able to view/click through to the producers, people, planets, vehicles, and starships that associated with any given film.
- The functionality (pages, PHP code, SQL queries) for submitting a new film for updating the database (e.g. add Episode 8: The Last Jedi). The fields to be included in the films (and in the HTML/PHP form) are to be based on the table containing the films and the tables linked to it. Yo should be able to post everything in one go, this will be multiple queries in the background — all one POST request.
- The SEO features/steps required for improving the ranking of your page on Google, whatever it is you deem possible to do for the on-site SEO

steps.

- The appearance and interactivity (the look and feel) of the website must use appropriate layout, graphics, and interaction flow, e.g.: make the images clickable to zoom-in/display a hi-res version to see the details. You may use suitable JavaScript/CSS libraries/frameworks to style and enhance your solution. For example, these are easy front-end frame works for designing: Bootstrap, Bulma, Materialize, Metro UI, Tailwind, etc.

For the advanced features you may choose to implement some or all of the following:

- Implementing login pages, and the features for creating user profiles/user registration, with appropriate features. This should use appropriate security precautions and avoid storing plain-text passwords in the database. There are two ways to approach/expand these features:

  1. The administrator users may only be able add or edit the content on the website.
  2. Users can rate films and add people, planets, starships, etc. as favourites. This will be visible via user profile. User might be able to view the favourites of other users.

- Implementing page(s) with graphs to display statistics of planets, people, vehicles, starships. Needless to say, it recommended that you use a JavaScript library for including/rendering graphs in your solution; for example, Chart.js (https://www.chartjs.org/) is a good choice for this project, but you may choose some other suitable library, or even write your own. Here's a page how to include Chart.js it in your project (https://dyclassroom.com/chartjs/chartjs-how-to-draw-bar-graph-using-data-from-mysql-table-and-php). Make at least five graphs to view any of the following, but not limited to these:

  1. The population of planets.
  2. Compare skin/eyes/gender of people.
  3. Compare size/speed/crew/passengers/credits of starships/vehicles.
  4. Whatever comes from your imagination!

- Make a Single-Page application using ReactJS library. The structure of the project will have to be changed quite a bit from the start, where PHP will be used a backend and ReactJS as frontend, in an ideal all-stars-solution scenario consuming a PHP REST API. This means that the PHP will be used to access the database and will provide the API that ReactJS will use to visualise/change the data. There's a bit of ReactJS material in the textbook, one chapter, but it is quite light on details — you will likely have to look into ReactJS documentation online to see the examples (only for the interface ReactJS usage): https://react.dev/learn, https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started.

Your solution must be accompanied by a four-to-five-page documentation/report presenting/describing the details of your solution. This documentation is essential to the graders — please include the information to help the grader gain a clear understanding of your project solution. *The reality of this is such that a weaker solution with a clear guide for the grader is likely to earn a better grade than an advanced solution with no grader guidance.*

Marking Schedule

The core tasks have 85% allocated to those, which is all the way up to the lower boundary of an A+, so to get into the A+ territory you'd want to think of the advanced features.

- Documentation/report — 25%. The documentation should be sufficient for testing and understanding your solution, presented in proper format and language.
- Basic features — 35%. The pages for showing the film, plant and people overview (1), click through to each part (2), edit and new data with forms (3) with appropriate navigation, structure, presentation.
- SEO features — 25%. The work demonstrating your understanding of the SEO techniques applicable to the scope and context of this project.
- Advanced features — 15% (each). You may choose to implement just one of those advanced features or all three — best solutions will be noted and acknowledged. And before you ask, no, you can't get more than 100% for this project.

Questions and Clarifications via Discussion Forum

Needless to say, you are invited to post questions via our Discussion Forum, and please be sure to discuss in the form anything related to this project, short of sharing your solutions.

Last modified: Friday, 8 September 2023, 12:50 PM

◄ Midsemester Test

Jump to...

Project Resources ►

?